AFRL-SR-AR-TR-03-

0011

REPORT DOCUMENTATION PAGE

OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
18 December 2002

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*
Jan. 1999 - Sept. 2002

4. TITLE AND SUBTITLE
Three Corner Satellite
Colorado Final Technical Report

5a. CONTRACT NUMBER

5b. GRANT NUMBER
F49620-99-1-0208

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Elaine Hansen,
Dave Beckwith,
Brian Egaas,
Steve Levin-Stankevich,
Jennifer Michels, Steve Wichman, Stephan Esterhuizen

5d. PROJECT NUMBER
1530941

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Colorado Space Grant Consortium
520 UCB
University of Colorado
Boulder, CO 80309

8. PERFORMING ORGANIZATION REPORT NUMBER
UCB-CSGC-02-005

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Jeff Ganley
Air Force Research Labs
3550 Aberdeen Ave SE
Kirtland AFB, NM 87117-5776

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/VSSV

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release;
distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
As part of the overall Three Corner Satellite (3CS) project within the University Satellite Program, the development and testing of the science Imaging System and the End-to-End Data System (EEDS) were undertaken at the University of Colorado in conjunction with team members from Arizona and New Mexico. This report describes the overall satellite project and the details of the development of the Imaging System and the EEDS.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	UL		19b. TELEPHONE NUMBER *(include area code)*

20030211 133

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 INTRODUCTION

1.1 Three Corner Satellite Description

The Three Corner Satellite (3CS) constellation is a cluster of three nanosatellites that are part of the U.S. Air Force University Nanosat program. The 3CS project was begun in January 1999 and the cluster of satellites is presently awaiting launch in 2003 from the Space Shuttle. The satellites are shown in their launch configuration in Figure 1. The 3CS project is a joint effort of the faculty, staff, and students at the three participating universities: Arizona State University (ASU), New Mexico State University (NMSU), and the University of Colorado at Boulder (CU). Consult [Unde 99], [Hans 99] and [Hora 99] for further details on the baseline design and mission concepts.

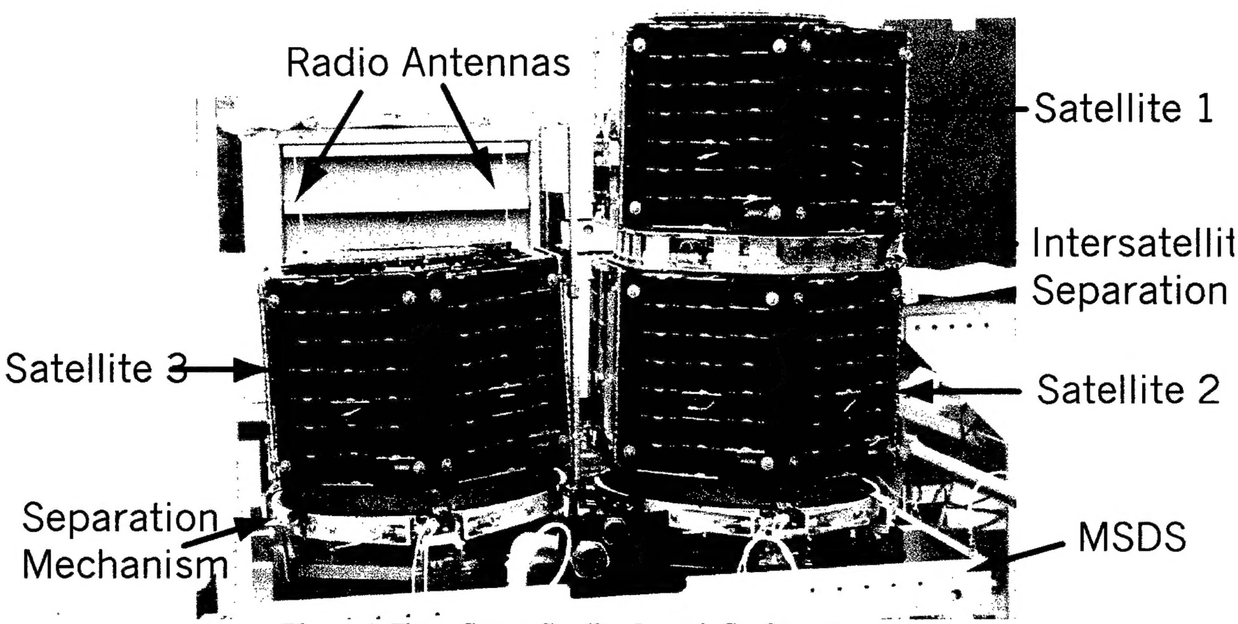


Figure 1. Three Corner Satellite Launch Configuration

The 3CS mission has four primary and three secondary objectives. The primary mission objectives include: stereo imaging, virtual formation flying, inter-satellite communications, and intelligent end-to-end command and data handling. The science will include imaging of clouds and other atmospheric structures using a formation of satellites. After deployment, the three satellites will operate together using formation flying techniques. This will be accomplished using virtual formation communications, a technology that allows the satellites to operate as a network utilizing communication and data links. Finally, the formation will use distributed and automated operations. This allows both individual nanosats and the entire formation to be controlled for optimum data gathering, command and control, and communication.

The secondary mission objectives include: validation of a MEMS heater chip for a Free Molecule Micro Resistojet (FMMR) propulsion system, demonstration of a generic nanosatellite bus design, and student education.

One of the principal features of the 3CS mission is that all three nanosats are based on the same design. Each university has responsibility for different subsystems, and all components are designed to be common to each of the satellites. Each school's team is comprised of a faculty member and graduate and undergraduate students. Each school leads in its respective areas of expertise and on strengths proven on past projects as listed below:

a. ASU – program management, structures, electrical power, micropropulsion, integration and testing.
b. CU – science (imaging), intelligent command and data handling, mission operations.
c. NMSU – communications.

Using this team concept, the design was developed and refined by the three schools. In addition to the design work, the team members verified that all components, materials, and design features would pass the NASA flight safety reviews for launch from the Space Shuttle. This report concentrates on the design of the satellites. Since the individual subsystems were produced by different partner members at each school, the overall set of components needed to be matched and integrated at final assembly.

The 3CS satellites were designed and fabricated as three common satellites. The only differences are antenna placement, the inclusion of the FMMR test electronics on two of the spacecraft, and some additional solar cells on one satellite. In this section, we will describe the common components for each satellite necessary for the mission goals. This includes the structure, the power system, the end-to-end data system, the imaging system, and the communications system.

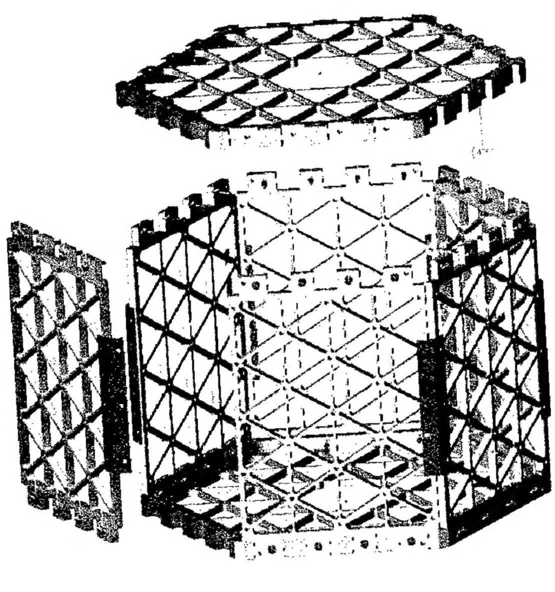


Figure 2. *Three Corner Satellite structure*

The structure for each satellite is based on a machined 6061-T6 aluminum isogrid structure as illustrated in Figure 2. The structure is hexagonal in cross section with the major axis for the satellite being 44.7 cm and the minor axis for the satellite is 39.4 cm. The height of each satellite is 29.2 cm. As can be seen in Figure 2, the side panels and the top and bottom plates form an interlocking structure. Brackets are added where the side panels abut for increased rigidity. The aluminum is anodized except for electrical contact points. The side and top panels are the load-bearing structure for the satellite. The structure was tested to verify that it would meet strength and vibration mode requirements for a Space Shuttle payload.

The Electrical Power System (EPS) is composed of the following elements:

a. solar cells,
b. battery pack,
c. inhibit relays,
d. DC/DC converter, and
e. microprocessor controller.

The system elements are organized as illustrated in Figure 3. The power from the battery and the solar cells is distributed as +5 V regulated and +12 V unregulated power. The 12-V power is used by the cameras and imaging electronics. The 5-V power is for other components. All systems but EPS have in-line switches controlled by EPS to prevent power-on prior to safe operating conditions being established after launch and to control the satellite power budget.

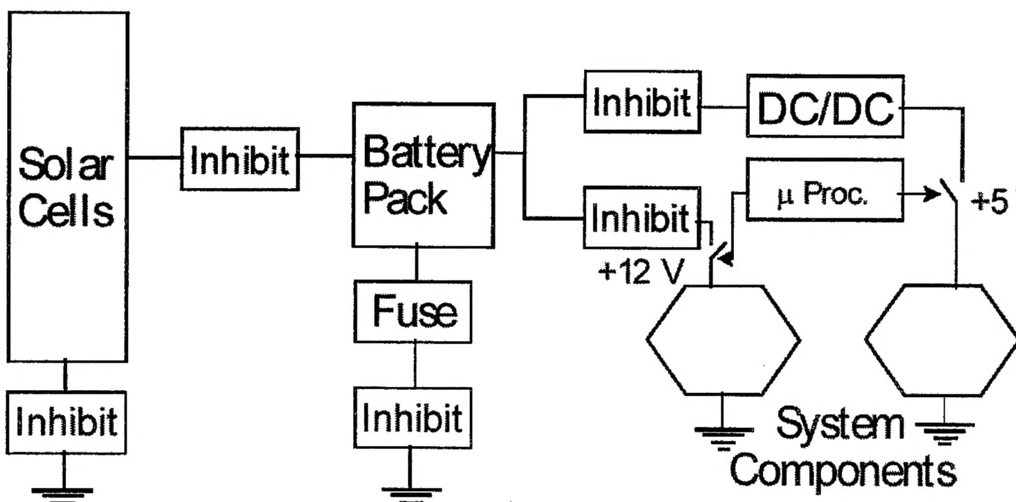


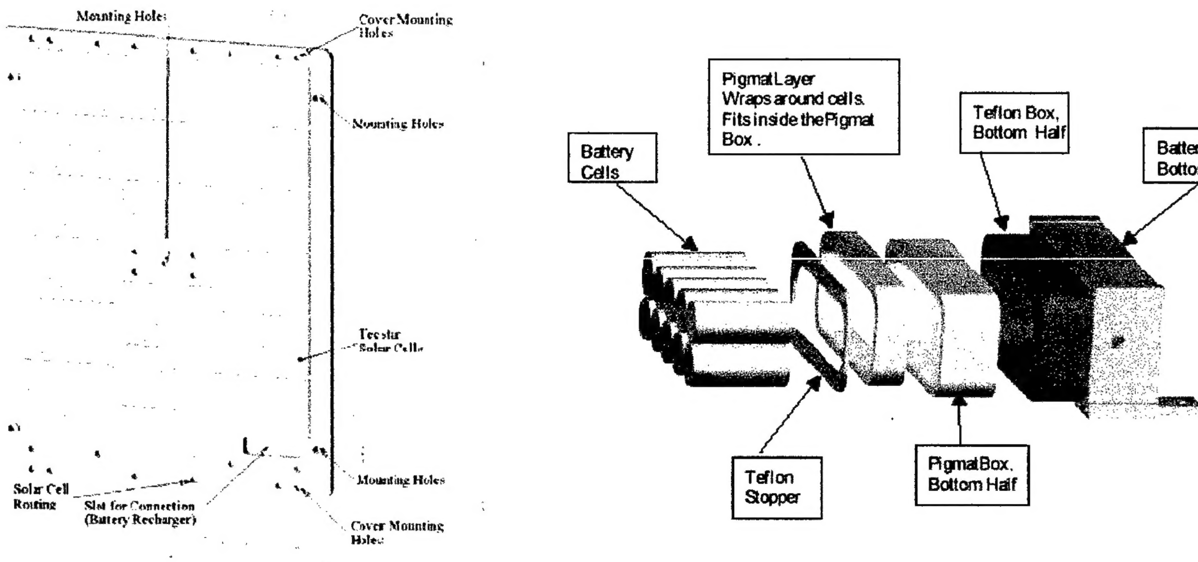
Figure 3. *3CS Power System*

The EPS microcontroller is a PIC microcontroller with a watchdog timer. The EPS microcontroller is responsible for

a. sampling and storing health and status telemetry measurements,
b. listening for end-to-end data system commands to enable/disable individual components, and
c. acting as a watchdog timer for the end-to-end data system.

The inhibit relays are required to prevent premature energizing of the electronics during launch. There are three latching relays between each power source and the load. There is also one latching relay in the ground leg for each power source. These relays are set by the inhibit timer electronics in the MSDS portion of the launch configuration.

The solar panels use dual-junction gallium arsenide solar cells that are commercially manufactured. The side panels and top panel of each satellite holds the solar panel structure in place. Each panel consists of two strings of nine cells with each cell generating approximately 2.4V. The solar panel contains a current meter and diode protection. The solar panel configuration is shown in the left half of Figure 4.



Special Side Composite Panel with Slot for Battery Recharging in the Stack

Figure 4. *3CS Solar Cell and Battery Configuration*

The battery portion of the EPS is composed of a pack of ten NiCd cells with 2300 mAh of capacity. The battery is illustrated in the right half of Figure 4 which shows the packaging into a vented box with absorbent material to mitigate any leakage effects.

The End-to-End Data System (EEDS) is an integration of computer hardware, software, procedures, and personnel for constellation command, control and data handling. The hardware component in each satellite includes an 823 Power PC flight computer having 16 MB of RAM, 8 MB of RAM organized as a solid state recorder, and a critical decoder for processing initialization commands.

The software is stored on disk and in ROM. The flight software is disabled until after deployment from the MSDS. The software uses a commercial operating system (VxWorks) and a number of commercial software tools for controlling the satellites. A primary software tool is the System Control Language (SCL) to serve as the control executive and to program rules to monitor and automatically react to the health and

welfare of the satellite. The current state of the satellite can initiate scripts to take counter measures for detected faults. The rules can also be used to generate scripts to perform specific functions under specific conditions, e.g. initialize the radio system or the imaging system. Operational science events such as taking a sequence of pictures are also controlled by SCL.

The command structure for the satellites is based on specific files sent from the ground stations, inter-satellite messages, and pre-stored, timed commands. The scheduling software can build a sequence of operational commands to be executed between ground station contact times.

The imaging system for the satellites is based on four cameras per satellite with the goal of producing images of clouds from orbit. There are two cameras on the top bulkhead and two cameras on the bottom bulkhead. The cameras interface with EEDS for data and control, and with EPS for power. The cameras are commercial cameras using a 640 x 480 pixel CMOS sensor. The cameras have full automatic exposure control. Onboard algorithms will be used to evaluate image quality and prioritize the images for downloading. The imaging cameras are illustrated in the right portion of Figure 5 and the mounting in the bulkhead is shown in the left portion of Figure 5.

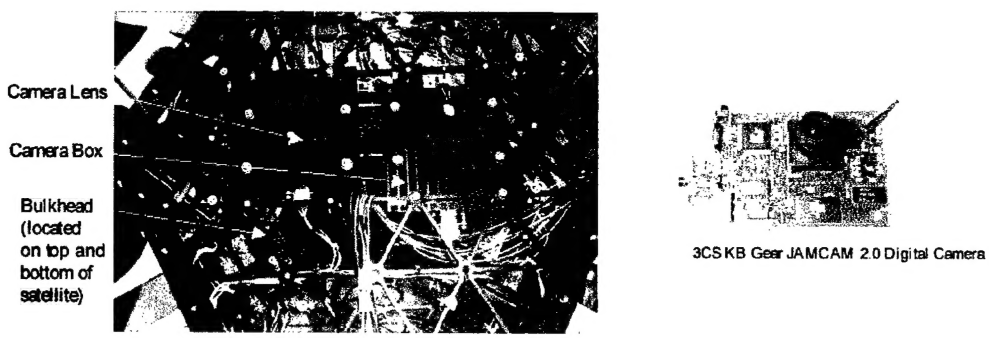


Figure 5. *3CS Imaging System*

The communications system is based on commercial amateur radio hardware that has been modified for an extended frequency range outside the amateur bands. The radios are arranged as dual redundant flight radios in each satellite. Each radio is software programmable via the EEDS to set frequency, power, and operating characteristics. The radios are only powered on when supplied power by the EPS to help control the overall satellite power budget with each radio being individually powered. The nominal operating mode will be to have one radio on at all times in receive mode to listen for commands. The radios use VHF frequency for the data downlink, UHF frequency for the command uplink, and a different UHF frequency for the intersatellite crosslink communications. All communications use a frequency shift keying technique and the AX.25 packet communications protocol at the physical channel level. The data rates available on the radios are 1200 bps and 9600 bps. The maximum transmission power is 1 W. The antenna system uses a commercial dual-band antenna for each radio. The antennas on the two satellites that are joined in the 2 plus 1 launch configuration have

their antennas inserted into a sheath inside the other satellite for launch. Upon satellite separation, the antennas are withdrawn from the sheaths without use of a deployment mechanism.

The primary ground control station will be at CU. The other schools (ASU and NMSU) will have compatible ground stations to be used as backup relay points for both commands and telemetry by using stored and forwarded data files in either direction. The uplink data files can be commands, schedules, or revised software. The relationship between the satellites and the ground stations is illustrated in Figure 6.

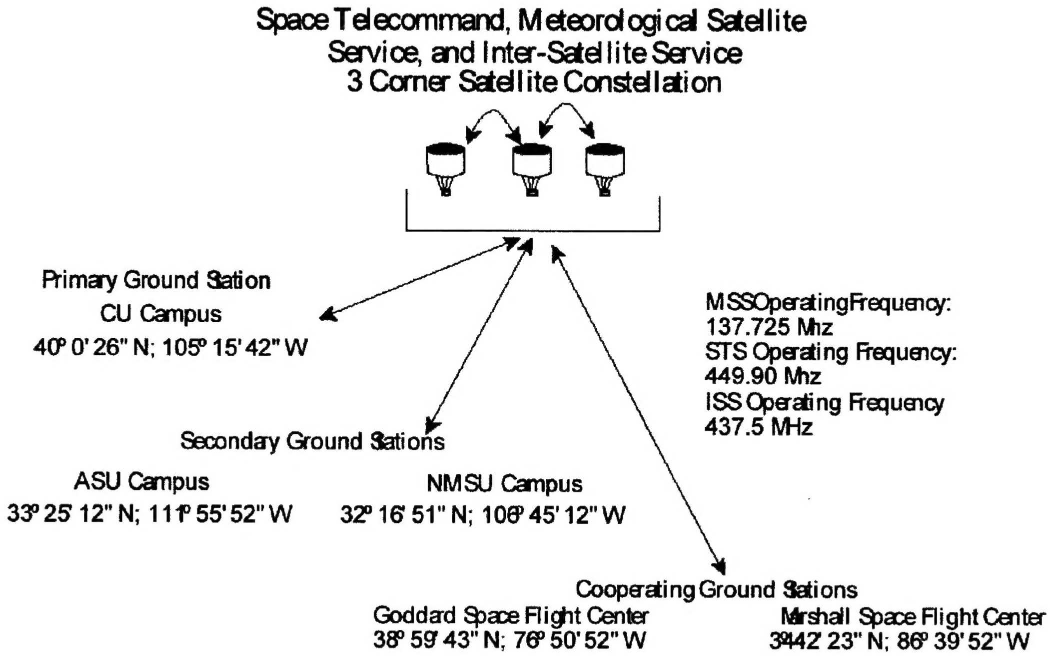


Figure 6. *3CS Communications Links*

The satellite cluster launch configuration was shown in Figure 1. The 2 plus 1 satellite configuration was needed in order to meet the vibration limits on the satellite separation systems. This configuration consists of

a. the individual satellites
b. the Lightband inter-satellite separation mechanism between the two joined satellites
c. The Starsys separation mechanism between the launch plate and the satellite on it
d. the Multiple Satellite Deployment System (MSDS) that holds the cluster configuration.

For the purposes of this project, the Lightband, Starsys, and MSDS components are "government furnished equipment' and outside the control of the 3CS project team. However, the satellites need to be compatible with these components. Additionally, the timer control and power for removing the inhibits are controlled through the MSDS

electronics. The electronic signals for activating the satellites are passed through the Starsys and Lightband components to the 3CS constellation members. This integrated system will be delivered to NASA to be mated with the NASA systems for launch.

1.2 Scope

The remainder of this report deals with the development of the 3CS science imaging and the end-to-end data systems that are part of the actual flight system.

1.3 Topic Development

The next sections contain the detailed description of the development of the science imaging system and the End-to-End Data System. We will discuss the design constraints and limits on developing the hardware, the testing to validate the components, and the production of the actual flight units. For the full details of the design, consult the documentation supplied with the fight units.

2.0 IMAGING SYSTEM DESIGN

2.1 Objectives:

Three Corner Satellite (3CS) has two primary science goals. The first is to image local atmospheric events in stereo with a range resolution of less than 500 meters. Stereo images of local events, such as cumulus towers, atmospheric waves, and aerosol plumes, will allow the assembly of three dimensional data sets including cloud shape and size. The second goal is to make a survey of cloud types, thickness, and altitudes. These surveys will assist with climate modeling and prediction by allowing us to create statistical maps of cloud types and properties. Surveys will consist of stereo imaging of clouds with spatial resolutions of approximately 1000 meters and range resolution of 500 meters. There are several advantages in using stereo imaging from space over conventional imaging. The first is to derive range data which can be substantially more accurate than range data acquired by other more traditional means, and also to cover a much greater area. The three nanosatellites will allow stereo imaging and the use of triangulation to determine accurate range data and to create three-dimensional images and depth maps.

To accomplish these goals the following objectives should be met.

a. Stereo image short-lived or highly variable atmospheric phenomena including clouds with their cumulus towers, waves, boundary layers and aerosols from dust storms, pollution zones, snow and ice.
b. Derive range data accurate to 500 meters, and
c. Create a sampling of cloud heights.

The resultant cloud range data will assist in:

a. Validating dynamic models
b. Modeling the global climate effects of clouds and aerosols.
c. understanding of local effects on regional aerosol loading, modeling, and climate change.

The science goals place a number of requirements on other 3CS systems. These include:

a. Pointing knowledge to 5 degrees
b. Temperature: -20C to +20C
c. Volume: 1500 CM^3
d. Mass: 1250 g
e. Peak Power: 3.5; Avg power of 2W
f. Peak Data Rate: 8.0 Mbytes/day

A primary scientific objective of the 3CS mission is to record stereoscopic images of short lived or highly dynamic (<1 minute) atmospheric phenomena (Figure 7), such as deep convective cloud towers, atmospheric waves, and sand/dust storms. Stereoscopic imaging is the simultaneous acquisition of images overlapping the same field of view from slightly different angles. By acquiring "stereo pairs" of the same scene, we can process the images to produce depth or range information, effectively measuring the

height of the atmospheric phenomena of interest. Cloud heights, for example, are critical to our understanding of the earth's climate and our ability to better model it. Using stereo imaging, we plan to measure the heights of clouds with a precision of ~1km and make a statistical study of their type, height, and thickness.

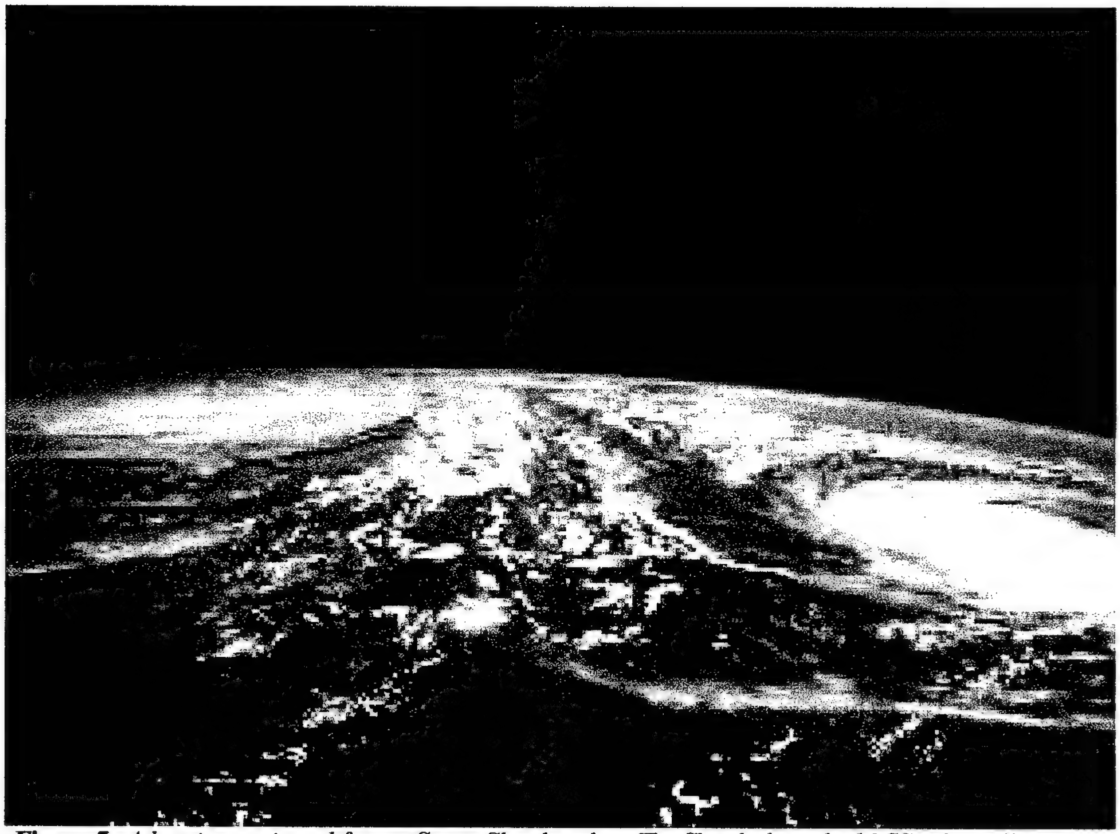


***Figure** 7. A hurricane viewed from a Space Shuttle orbit. The Shuttle-launched 3CS orbit will provide an excellent vantage point from which to acquire images of clouds and other atmospheric phenomena.*

2.2 Imaging Concept

The imaging subsystem onboard the 3CS constellation provides the capability to take digital still photos of the earth and its atmosphere from orbit. Each of the three satellites has four color cameras pointing in different directions so that the earth is always in view through at least one. Using crosslink communication, the flight computers on each spacecraft can coordinate the acquisition of images to provide the stereo pairs necessary to compute clouds heights. Once downlinked, these image pairs will be processed to produce maps of cloud heights (Figure 8). This image processing requires that the two images first be registered to each other according to background features, such as land and water. The apparent shift in position of cloud features is then directly proportional to their height, which can then be computed.

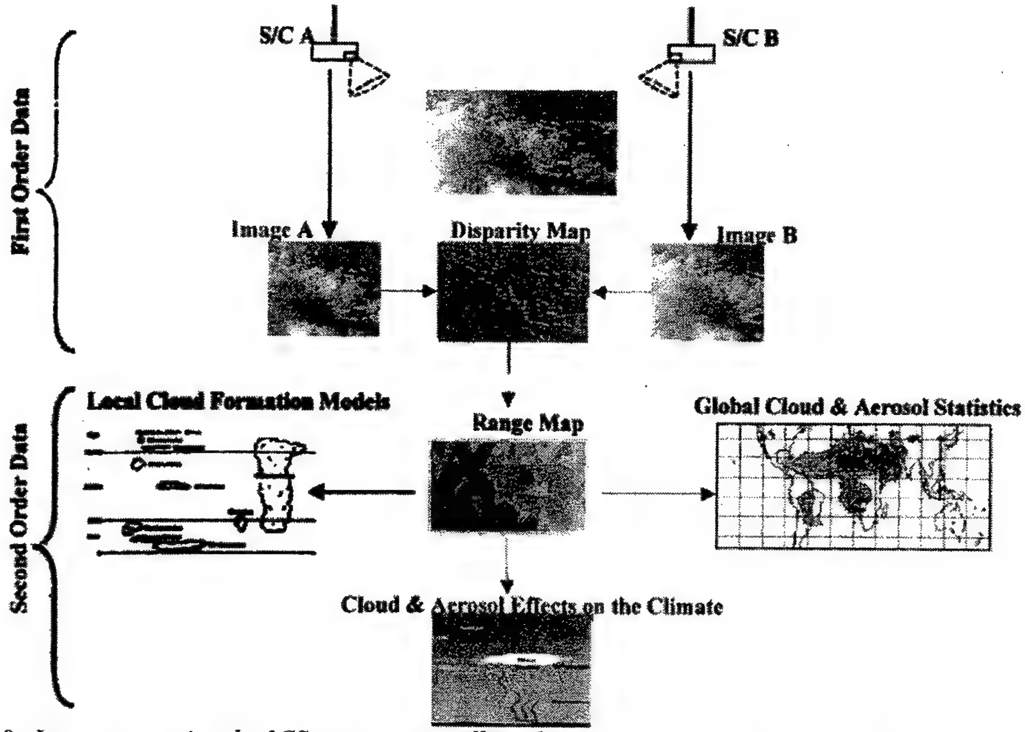


Figure 8. *Image processing the 3CS stereo pairs will produce measurements of cloud heights throughout a scene.*

2.3 Imaging Operations

The 3CS flight computers and software control the scheduling of image acquisition, onboard processing, and downlink of all image data. When two spacecraft are in crosslink range, a stereo pair of images can be acquired simultaneously by two separate spacecraft. In another mode, regardless of spacecraft location relative to the other satellites, a quasi-stereo image pair can be acquired by a single spacecraft by taking one image, waiting a short period (<30 seconds), then taking another. The software that runs the imaging system is described elsewhere in this report.

2.4 Imaging Hardware Design

The 3CS cameras are modified versions of off-the-shelf, hand-held cameras manufactured by KB Gear Interactive (Figure 9). These cameras are built around HP/Agilent CMOS sensors (Figure 10), an emerging technology that's cheaper and uses less power than traditional CCD devices. The resolution is 640x480 pixels, with full color from a Bayer masking array. The field of view is 43° x 32°, which gives approximately 0.5 km nadir resolution at our orbital altitude. Fixed aluminum/glass lenses are mounted in a polycarbonate lens holder attached to the camera circuit board. Cameras are modified by replacing electrolytic capacitors, removing extraneous jacks and buttons, and wiring the necessary signal lines.

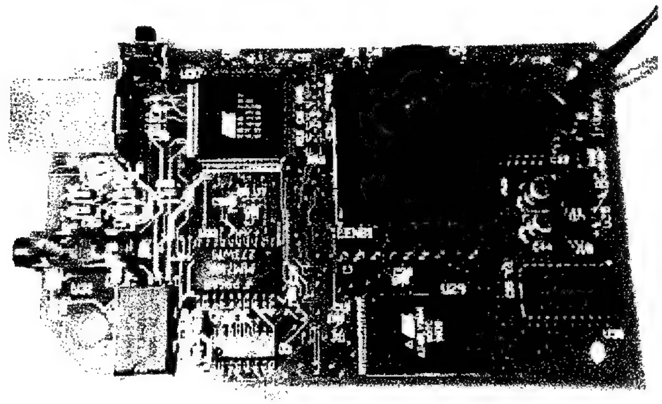


Figure 9. *The single board camera from KB Gear met the requirements for the 3CS imaging system.*

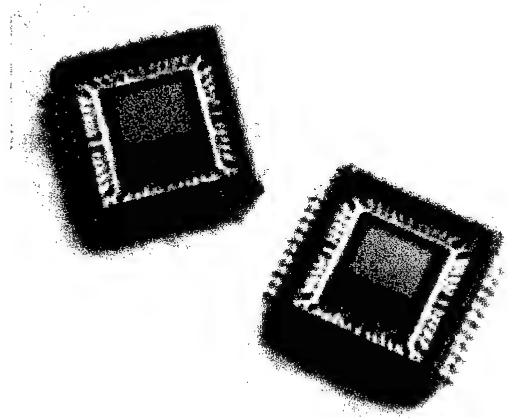


Figure 10. *CMOS sensors are cheaper and use less power than comparable CCD devices.*

These cameras were chosen because they met the needs of the 3CS development in a variety of ways. First, with their small, single board design, no moving parts and large mounting holes, they were rugged enough to survive the stringent environmental testing required of the 3CS components. Second, their specifications were sufficient for the proposed imaging mission. Third, they were both available and inexpensive, allowing us to purchase over 50 units for development, flight, and SisterSat construction. But perhaps most importantly, we were fortunate that manufacturer KB Gear Interactive was very generous in providing specifications and instructions on the serial commands necessary to operate the cameras, which cut development time considerably. The cameras are controlled by the flight computer over RS-232 serial lines. Additionally, I/O lines for turning the cameras on/off, activating the electronic shutter, and selecting which camera to use are utilized by the flight computer. A camera interface circuit was designed and built at CU to switch control and data signals between any of the four cameras and the EEDS flight computer, as well as provide power from the EPS subsystem (Figure 11).

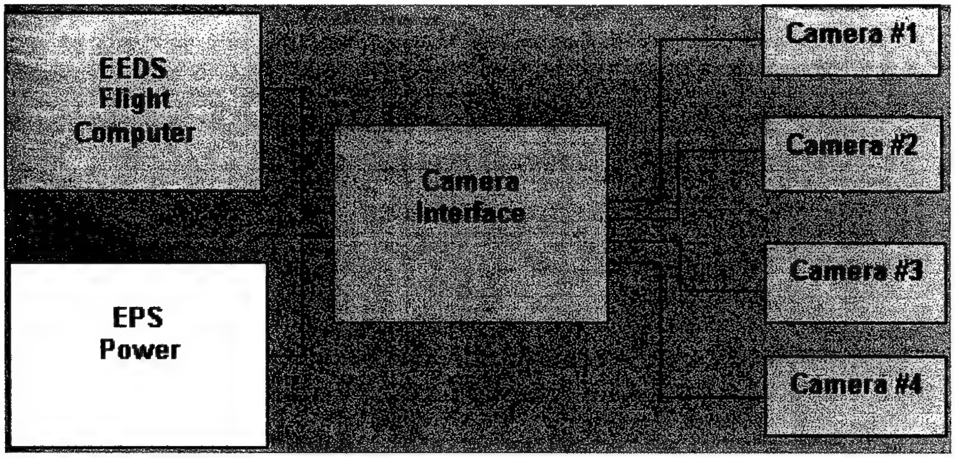


Figure 11. A camera interface board routes all electrical lines to and from the cameras.

Cameras are mounted on angled brackets inside anodized enclosures attached to the spacecraft upper and lower bulkheads (Figure 12). The boxes for both the cameras and the camera interface were designed by ASU.

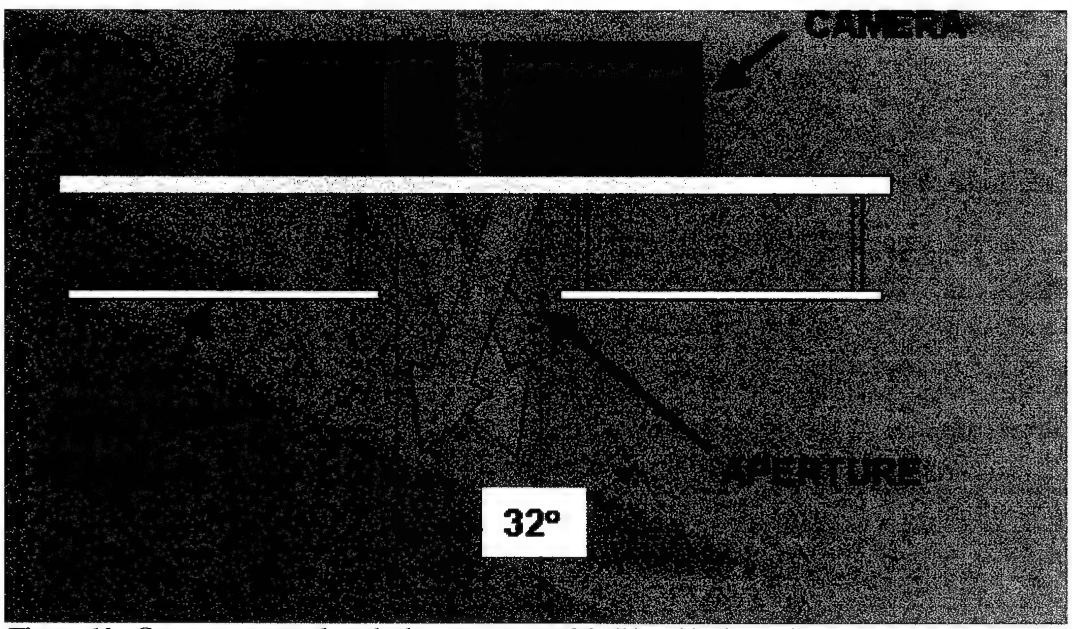


Figure 12. Cameras mounted on the lower spacecraft bulkhead look out through a single hexagonal aperture in the isogrid structure. The mounting brackets are tilted so that the two cameras provide contiguous adjacent fields of view.

2.5 Imaging System Testing

Formal testing of the imaging subsystem hardware is done as part of the EEDS functional test, since the cameras are not operated independently of the flight computer. During the functional test, each camera is activated to take and transfer a picture, which verifies that the control and serial data lines are all connected and operational. Test problems during

development centered around minor software errors and improper electrical grounding in the testbed. The imaging subsystem utilizes the ground from EPS. However, problems can arise if the signals from the flight computer are judged against the EPS ground and EPS and EEDS don't share a common ground, which occurred in simulating spacecraft power with separate power supplies. All problems have been resolved, and we presently have 12 fully functional cameras onboard the three spacecraft, ready to carry out the 3CS imaging mission.

A proof-of-concept experiment was conducted to test our ability to measure object heights in a laboratory scale (1 cm in the lab : 1 km in the real 3CS mission). We attached a 20 cm piece of foam to a wall to represent a 20 km tall cloud tower, and positioned a camera 350 cm from the wall to represent an orbital altitude of 350 km. While maintaining this distance between camera and wall, we took pairs of images with the camera in two positions, simply shifted laterally. By measuring the shift of the foam "cloud" relative to the background in the resulting images, we computed the object height. As the graph below shows, we accurately determined the object height with increasing precision as the separation in camera position between the two views increased (Figure 13). It will be more difficult to match features in real imagery from orbit, but this is a good test of the principles involved.

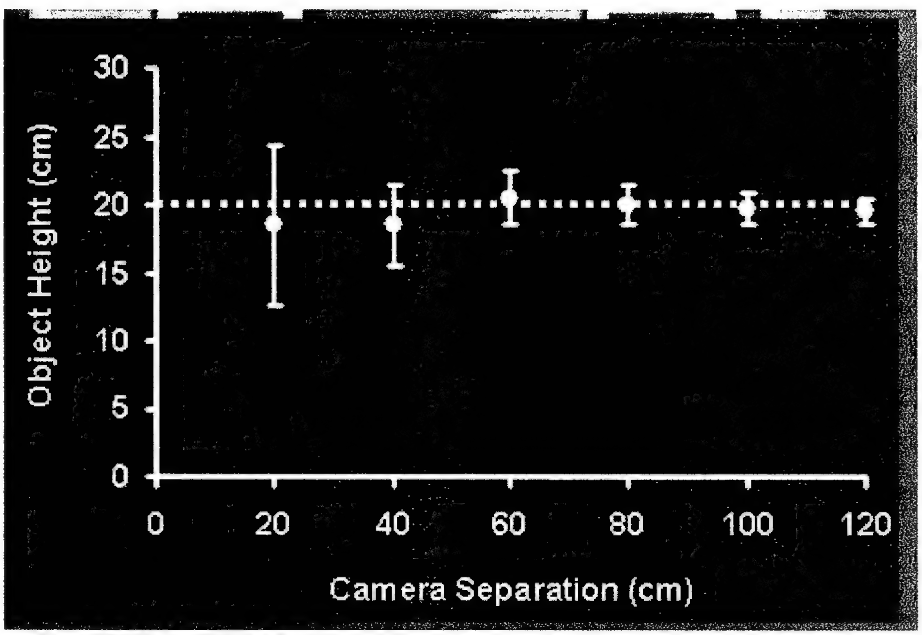


Figure 13. *Stereo imaging allowed us to accurately measure an object in a scaled experiment in the lab.*

2.6 Imaging Mission Constraints

Two major system constraints limit the capability of the 3CS imaging mission. First, data communication rates will severely limit the amount of imagery downlinked to the ground systems. This is further exacerbated by the short orbital lifetime. Second, the lack of knowledge or control of spacecraft orientation will make the acquisition of good stereo pairs of images difficult, though not impossible.

2.7 Significant Achievements

Simply getting the imaging system hardware built and tested is our most significant achievement. There were serious discussions of abandoning the imaging mission at some points during the development. Finding and acquiring suitable cameras proved difficult. The current camera is the third camera selected, as the first one and then a second became unavailable during the development. A mechanically complex camera pointing system was eliminated to simplify the design. However, we also upgraded from the 5 cameras originally planned to the current complement of 12 to compensate for the elimination of the ADCS from the 3CS design. In the end, we have more cameras, with full color, and higher spatial resolution than originally specified.

Second, the scaled stereo imaging experiment in the lab demonstrated that the imaging system we designed is, in principle, capable of fulfilling the 3CS stereo imaging mission.

2.8 Imaging Software Detailed Design

Once the camera hardware was integrated with the software, there were two tasks that needed to be done. The first was to convert the image format that the camera uses to a JPEG. The second was then to rank the image on a scale so that mission operators had an indication whether they wanted to spend the time to downlink that particular image.

2.8.1 Image Format Conversion

The JamCam 2.0 cameras have been completely integrated into the 3CS system. The system uses them by running a routine of serial commands to ping the camera, clear the memory, take a picture, download the image from the cameras to the flight computer, and then process the image. The image has to be processed from an 8 bit raw format to a jpeg format. This was done by using an interpolation technique to convert the image to a 24 bit format, and then compressed using a jpeg algorithm. The images are anywhere from 10K to 50K. The shutter speed on the cameras is a limiting factor in the mission. Since this shutter speed is relatively low, blurred images could result if the satellite is rotating too fast. Another limiting factor with the camera system is that since the satellites are tumbling, it is unlikely that the cameras on all three satellites will be pointing at the same thing. An algorithm is needed to determine the rotation rate of the satellites so that mission operators can predict which cameras need to be used to have cameras on all three satellites point at the same thing. An onboard algorithm is also still needed to process a set of stereo images and determine the height of an object in the images.

2.8.2 Image ranking

All images acquired by the spacecraft are scored by an image ranking algorithm to prioritize downlinking the best images for our stereo imaging task. This simple algorithm categorizes each pixel according to its color and intensity as either the blackness of space, the gray of clouds, or the varied terrestrial background of water and land. For our stereo

imaging mission, we desire images which have both clouds and background, and as little black space as possible. The onboard algorithm computes a score for each image, in the range 0 to 100 (100 is best), crediting for the presence of cloud and background pixels and penalizing for the presence of black space. High ranking images are given priority for downlink, although we retain the ability to manually override this score for any image.

3.0 END-TO-END DATA SYSTEM DESIGN

The EEDS detailed design can be broken down into three areas: hardware, CSGC flight software (FSW), and COTS software. All three of these areas designed simultaneously leading to the completed 3CS design.

3.1 EEDS Flight Hardware

The End-to-End Data System hardware is comprised of two flight components. These components are an embedded system and board that interfaces the embedded system with the rest of the satellite's subsystems. When the design originally started, the objectives of the hardware were to build a system that could control the satellite, have enough memory to store pictures, have enough memory and speed to use JPL's test software, and have the whole system run on 5W or less. To accomplish this task, an RPX Lite was selected from a company called Embedded Planet. The RPX Lite includes a Motorola Power PC 823, 16MB of RAM, 16MB of Flash, an Ethernet port and three serial ports. The interface board was designed by students at CU. The interface board allows two subsystems to share a serial port due to the fact that the satellite needs four and the RPX Lite only has three. The interface board changes the voltage levels of one of the serial ports so it is compatible with the other systems. It has a watchdog in case the computer hangs and needs restarting. The interface board also aids in the control of the imaging system. Below is a flow diagram of the EEDS hardware system.

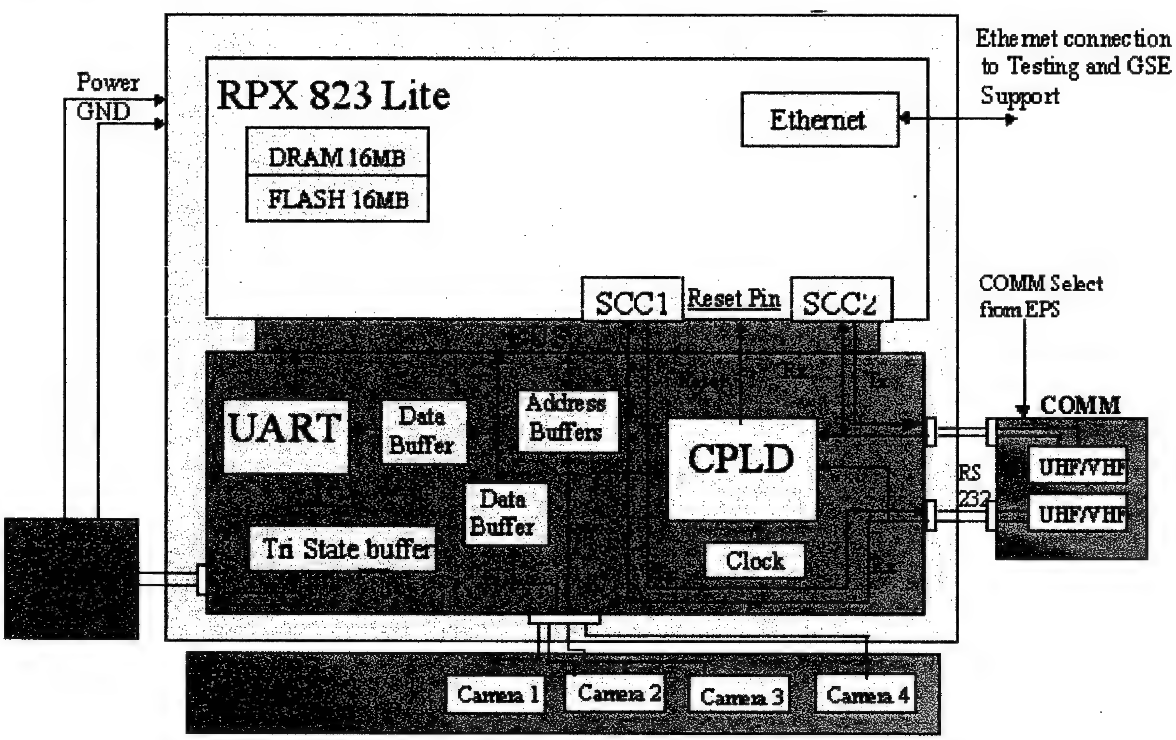


Fig 14. EEDS Hardware

Testing started by trying to get the interface board and the flight computer to talk to each other and send out the correct signals. Below is a picture of the some of the

initial testing done on the boards. All of the wires hooked to the boards are also hooked to an digital logic analyzer that was used to determine if the signals on the serial ports were the correct signals. This testing showed several flaws in the original design of the interface board but all of these were corrected in the second revision of the board.

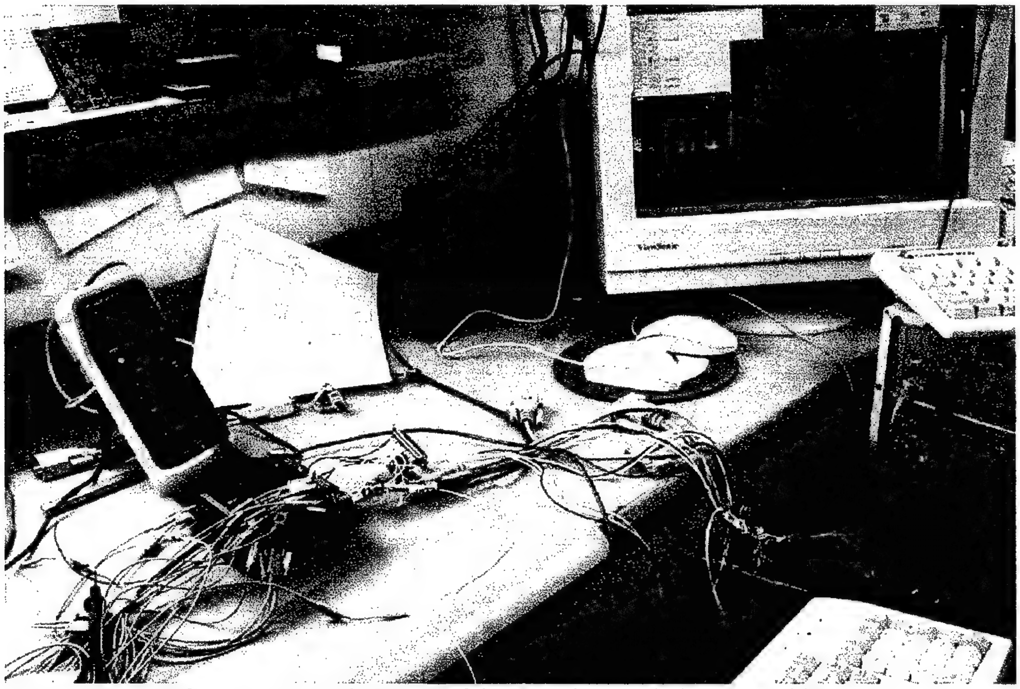


Figure 15. *Initial Testing of the Boards revealed some problems.*

Once both of the boards were working together, the Electrical Power System, EPS, was integrated with the boards. Testing started by getting both systems to work together. This was also a big test of the software being used as well. All of the subsystems were integrated to EEDS and then tested. The Ethernet port became invaluable during all of this testing. The Ethernet port provides the tester with an easy and convenient way to log into the flight computer and control it. Later on when all of the subsystems were integrated, this provided an easy way to remotely test and control the satellite. This was very important in this project due to the fact that integration was occurring at Arizona State University while software development was occurring at the University of Colorado.

The following table details the specifications of the EEDS Flight Hardware:

Processor	MPC823
DRAM	16MB
FLASH	16MB
NVRAM	N/A
10Base-T Ethernet	CSS2-10Base-T (RJ45)
Monitor Port	SMC 1-3 wire RS232 (RJ45)
Serial EEPROM	12C
Serial Temp and Thermal Monitor	12C

Debug	Development Port header for BMD
TAP	TAP header for test and JTAG
PCMCIA	Single Slot – Type I, II, or III
Dip Switch	4 position slide switch read via status register
LEDs	Two user programmable via control register
Bus Expansion Receptacle I	Processor Bus Interface Expansion Receptacle
I/O Expansion Receptacle I	Processor I/O Interface Expansion Receptacle

***Table 1**: FC Technical Specifications*

The EEDS Interface Board (IB) provides communications between the flight computer (FC) and the Electrical Power System (EPS), Communications system (COMM), and Imaging (IMG) System. The IB also handles the critical decoding for the ground-reset command, the watchdog reset, and the serial switch. All functionality of the interface board remains transparent to the operator. The block diagram of the EEDS flight system hardware is included in Figure 14.

3.2 Flight Software

Interfacing the EEDS hardware with the software onboard was done through custom CSGC flight software (FSW). Interfaces were made to the cameras, to the external power supply (EPS), and to the radios as described below.

3.2.1 Imaging Software

To interface to the JamCam 2.0 cameras documentation was acquired from the KBGear Company. They provided us with a document which stated all the serial commands needed to take a picture, clear the camera memory, download the image, and ping the camera. These serial commands were in the form of a 4 character string with extra parameters as needed. To use these commands, software was written that would access the cameras over the serial port and send the appropriate 4 character string for a given command. Before taking a picture the memory is cleared of old pictures and then a picture can be taken to avoid confusion of which picture is which.

3.2.2 External Power Module Software

The External Power Module (EPM) is the interface between the FSW and the EPS. It was based on the Interface Control Document (ICD) software from ASU. This ICD defined the two byte header which precedes all EPS commands, the command byte which is understood by the hardware, and the footer byte which indicates the end of the command structure. A FSW command was given for each EPS hardware command that was recognized. The user would use the appropriate FSW command to access the EPS hardware. The limitations of this system are speed and the concern that untested commands could get to the system.

3.2.3 Communications Software

The FSW has implemented its own communications protocol for transferring files and commands between each of the satellites and the ground. The COMM system receives and creates several different types of files during normal operation. The different types

of files are commands, image files, schedule files, software updates, and health and status files. Commands are simple, one frame files. The protocol for a command is to send the frame and then wait for an acknowledgement that it was received. If the command is not acknowledged within a certain period of time, a retransmission of the same command is attempted up to 20 times. Image files are created on the satellites by the satellite's local camera system, and received by the master satellite from the slave satellites. These image files are stored in /tffs3 on the satellites and can be displayed in the image catalog. Schedule files are sent from the ground to the master satellite in a compact form usually around 5K in size. Once the CASPER module on the master satellite begins running the schedule, the compact schedule is un-parsed and can reach sizes of up to 50K-60K. This is a limiting factor when changing masters because the satellites have to transmit the large file used by CASPER. Software updates are files sent from the ground to the satellites in orbit. These files contain code that will change the flight software on the satellites. The procedure for using and sending these types of files is still incomplete. Health and status files are transmitted from the slave satellites to the master satellite. These files contain telemetry about the battery levels, currents, temperatures, and the state of the software. When these files are received, they are stored in an SCL database that corresponds to the satellite number that the data belongs to. The master then sends telemetry saved in its databases about all three satellites to the ground. As of December 16, 2002, the unresolved problems include:

a. Speed
b. Possible pitfalls
c. Crossband, and
d. Others?

3.2.4 Software Manager

The main function of the Software Manager (SWM) is to get the satellite's Health and Status (H&S) data. This function queries several parameters of the software and reports them back in the broadcast packet. Parameters such as which tasks are running and which are stalled are essential for mission operators and onboard safety rules to determine the state of the satellite. If any of the essential tasks are not running, the satellite will automatically reboot itself in an attempt to fix the problem. Other parameters such as number of images waiting for downlink, the amount of disk space available, the amount of RAM available, and the communication status are also reported.

One of the main features of our EEDS development is to allow the flight software to be updated easily while in flight. SWM is the FSW module responsible for this operation. During flight, the mission controllers may find that software can be improved (to get more or better science data, for example). The FSW has a modular design such that most pieces can be uploaded and installed. The COTS software, CASPER and SCL, however, do not follow this design. These executables are too large to upload over our ground-to-space communications link. In addition, there are supporting files for both CASPER and SCL, however, that can be uploaded and we plan on updating these files frequently.

As of now, FSW software updates can be accomplished with a single command. However, there is an added security feature that we are working on to safeguard this

entire process. When the mission operators upload the new change, the simple procedure would replace the old code with the new code and reboot the satellite. However, if the new software gets corrupted in any way, erasing the old (or even moving it to a different place) might jeopardize the mission. This piece of software could be essential to future communication with the satellite. Therefore, we have proposed that a command, cmd_swm_test_update, be used to load the new code from a temporary file location. This way, if the software is corrupt, or we can't communicate with the satellite for a long period, the satellite can reboot itself and come back up with the old software still intact. The FSW test of new software seems complete, but uploading new supporting files for SCL (COTS software) is still under development.

3.2.5 Broadcast Packet Generator Software

The Broadcast Packet Generator (BPGEN) controls…

a. Byte stuffing

b. Packet generation

c. Limitations

- Speed
- Possible pitfalls
- Others?

d. Unresolved problems

3.3 Commercial-Off-the-Shelf Software

The COTS software used onboard consists of two products: CASPER and SCL.

3.3.1 Continuous Activity Scheduling Planning Execution and Replanning (CASPER)

The Continuous Activity Scheduling Planning Execution and Replanning (CASPER) software was designed at the Jet Propulsion Laboratory (JPL) by the Artificial Intelligence Group. The interface between CASPER and the rest of the satellite is through the SCL Software Bus. This is a publisher/subscriber interface where SCL will listen for messages posted on the bus by CASPER. CASPER will command SCL to execute goals and receive periodic updates as to the state of the spacecraft and goal execution. The information coming back from the FSW will be placed in the SCL database where CASPER can get that information.

The main supporting files that CASPER uses are the spacecraft model files and the daily schedule file. The model files tell CASPER how much of which resources all the particular goals require and all the restrictions on performing these goals. The schedule file will be uploaded by the mission operators about once a day. This file contains the goals that the mission operators would like the satellite to perform during that period. With this information, CASPER can take the schedule file and apply the model rules it knows. If a particular goal at a particular time does not have adequate resources, then CASPER can elect to move the activity to a new time that does have the right amount of resources, or it can delete the goal.

As the health and status data is updated in the SCL database, CASPER will watch the numbers that get put there. If the resources reported are unexpected, CASPER will reevaluate the schedule it originally proposed and shift goals around to satisfy mission needs.

The main limitations to the CASPER found so far are the schedule size and the time it takes to create and repair the schedule. With the slow bandwidth of the COMM system, we have to be aware that the size of the schedule file, after being established the first time by CASPER, is quite large. The original file of goals from mission operators is small in comparison; however, CASPER adds a lot of information to that file. If the master changes from one satellite to another, it is this large schedule file that must be transferred across the intersatellite link. The second limitation is the processor speed. While creating the schedule for the first time, CASPER must maintain efficient and smart heuristics that will repair the schedule as fast as possible. If the initial schedule of goals from mission operators is too large, the processor could take many hours of computation. As of now, a 24 hour schedule of goals takes between 10 and 20 minutes to repair which has been determined to be satisfactory.

3.3.2 Spacecraft Command Language

Spacecraft Command Language (SCL) was designed by Interface Control Systems, Inc. The advantage of using SCL as an off-the-shelf product was the integrated SCL language. This language provides the means to remove the mission operators from knowing all the detailed FSW commands and to provide them with simpler commands to accomplish goals and maintain the spacecraft's health.

An advantage of using a scripting language such as SCL is the ability to run entire goals from several or even one script. The goals are a series of commands that have the timing and error checking built in. Some goals of 3CS are simply executed on one satellite. In this case the goal often only runs off of one script containing many flight software commands. These scripts may also include logic to ensure the satellite is in the correct operating state prior to executing the activity. The script then sends out each command to the hardware in the correct sequence and updates the satellite's state accordingly. In the case of multi-satellite goals, such as stereo images, the scripting process is more complex, yet similar. Before each satellite runs the goal through a script onboard, it first must be told what activity to run. Therefore, a script is executed on the master satellite to schedule the activity on itself and the two slave satellites. Once these transmissions have been completed it is up to each individual satellite to run the scheduled script, thus competing the activity or goal.

A major aspect of autonomously operating the satellite is ensuring its health during flight. For this 3CS utilizes SCL rules. SCL rules allow for monitoring the satellite's state and reacting to detected faults. As 3CS is not a very complex satellite, having only a few subsystems, the rules that were implemented were fairly basic. In most cases, if a voltage or current sensor is detected as being out of range it will be logged to a file that we may download later. Some pieces of hardware, such as the cameras, may be turned off if their voltage sensor is out of range. Rules are used to monitor the battery level and ensure that we don't perform activities when the battery is below nominal levels. Other rules monitor several sensors and set the satellite to different states. These states may then be viewed on

the ground to determine the overall health and status of the satellite. These 3CS rules provide a general safety net to safeguard the satellite during autonomous flight.

SCL has a list of commands it recognizes as commands to pass on to the FSW. This is how SCL scripts and rules perform all the normal daily operations of the spacecraft. The FSW commands are issued to the FSW and SCL waits for a return code. SCL will wait a specified amount of time for this return code. If it is not received in time, SCL assumes there was a problem and follows the code to determine how to handle the situation. When the return code is received, this value is checked as to the appropriate value expected. If the value is not appropriate (i.e. there was an error), then the SCL scripts will know how to deal with this circumstance.

The FC boots VxWorks, a Realtime Operating System (RTOS). This RTOS allows for easy creation and integration of code and ensures that all tasks will execute within a certain time.

3.4 EEDS Operations

One of the main responsibilities of the University of Colorado is developing the mission operations system for the 3CS project. This is accomplished through the use of several pieces of software. Figure 16 illustrates the software and the flow of the mission operations data for the mission.

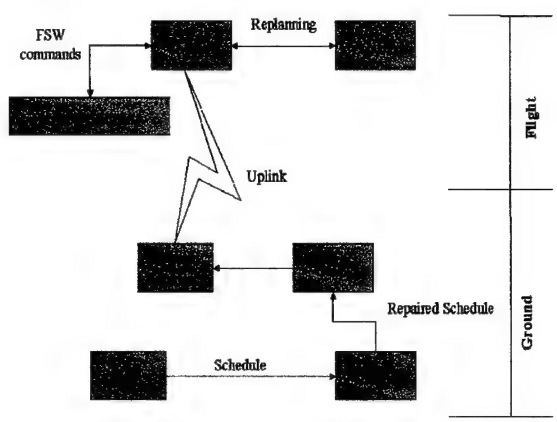


Figure 16. *Flow of mission operations system on flight and ground.*

Mission operations for 3CS starts with the use of Satellite Tool Kit, STK. Based on our orbit, STK generates timeline reports. The reports include when to run activities as well as ground station pass times and orbital events such as day/night times.

The reports are formatted so that the ASPEN software reads them as input. The Automated Scheduling and Planning Environment, ASPEN, software creates a schedule and repairs any conflicts in the schedule. A conflict in ASPEN, for example, is when the battery is estimated to drain below nominal or two activities are scheduled to overlap. While ASPEN has the ability to resolve these conflicts, it will not be used to completely repair the schedule. ASPEN will be used to check the schedule for major conflicts and invalid formats. The reason ASPEN will not be used for repairing the schedule is that CASPER will repair the schedule onboard the satellite.

CASPER (Continuous Automated Scheduling Planning Execution and Replanning) is innovative new software, developed by researchers at the Jet Propulsion Laboratories (JPL). CASPER uses the schedule checked by ASPEN to plan a time period. If the schedule doesn't have any conflicts, the activities will be run successively. However, if a conflict arises and a scheduled activity cannot be met, CASPER has the ability to rearrange the entire schedule to meet the missed objective. CASPER is continuously analyzing and modifying the schedule to meet the mission goals and run most efficiently.

The common link between the schedule on the ground and in flight is SCL. SCL is high-level control software used both on the ground and in flight. SCL will be used as a scripting language for flight software commands onboard and for mission operations on the ground. The flight software is used in flight to control and monitor the satellite subsystems. SCL provides the link between CASPER and the satellite subsystems.

The design of 3CS's mission operations system is based upon providing autonomous mission operations. The mission operations protocol provides a framework for autonomous control over the satellites. Once the user creates a timeline report using STK, the rest is done without input from the operator. ASPEN will analyze the files output by STK. SCL will use a script to send the schedule to the satellite during a ground pass. CASPER will then pick up the schedule and begin running the activities. The activities will again use SCL to control and monitor the flight software. The only user input comes at the level of creating a timeline of activities for STK.

3.4.1 Operations with SCL

SCL is a high-level control tool that provides partial autonomy to the satellite. SCL is a COTS (commercial off the shelf) package that can be tailored to a specific project through a database, scripts, and rules. The database consists of many object classes including measurements, actuator states, and derived data. These values are then accessible in scripts and rules that utilize the SCL programming language. The language allows logic based programming such as C++. SCL scripts provide a series of commands to execute a given activity. Rules provide a method for analyzing health and status values and monitoring the system.

Together, scripts, rules, and the database can be used to provide flight autonomy. For example, a derived database item can be monitored and may set a rule to fire, which in turn will start a script that changes a state on the satellite and provides for safe and efficient operations.

3.4.2 Constellation Operations

Three Corner Satellite will be flying as a constellation of three satellites. The constellation will allow the science team to meet their objectives through distributed tasking. Constellation flight requires a method for inter-satellite communication, known as the Inter-satellite link (ISL). Several factors arise when planning a mission involving a constellation of satellites such as: How do the satellites interact in space? How does the ground command a

constellation of close ranging satellites? How does the ground analyze the constellation's performance and health?

Master Control

The method chosen to control the constellation most efficiently is known as a master-slave relationship. One satellite will be dubbed the master and will maintain control over the entire constellation. The complex part of the ISL deals with how the master will control the slaves without losing any capabilities in the slaves. The first part of the discussion regarding control focuses around the use of the software on the satellites.

Three main components of flight software reside on the satellites. The actual c-coded flight software (FSW) that talks to the hardware, SCL, and the scheduling software, CASPER. The FSW is required on all three satellites to send commands to the hardware. However, is it necessary for both SCL and CASPER to run on all three satellites?

Initially, ISLs were used to send a series of FSW commands from the master to the slaves, thereby eliminating a need for SCL to run on the slave satellites. The slave would then receive each command individually and perform the requested activity. This idea was proven inefficient because individual commands could be lost in transmission and an entire activity would have to be re-started or ignored altogether.

The final plan for ISL requires SCL to be run on all three satellites. When the master wishes for the slave to run an activity, it sends a command package including a script id and up to several parameters. The script id is received by the FSW on the slave satellite and it executes a SCL script with the parameters specified. For example, when a stereo picture is scheduled for the constellation, the master sends the command string "1018 'time' 0" to the slave satellite. 1018 is the record id of the script to take a picture on the satellite, 'time' would be a string indicating a UTC time for when the picture should be taken, and 0 is the camera number ranging between 0 and 3 for the four cameras onboard each satellite.

The last question was whether to run CASPER on all three satellites or singly onboard the master satellite. It was decided to allow only the master to run the high level scheduling software, CASPER. This is most efficient because the constellation's activity may then be scheduled entirely through the master satellite.

Ground Commanding

Controlling a constellation of satellites from the ground is difficult for a couple of reasons. First, the constellation will be in a low-earth orbit, below the International Space Station, and therefore will have short ground station passes, on the average of 6 or 7 minutes. Second, the communication system is based on HAM radio operation with all three satellites on the same frequencies. It is, therefore, not efficient to try and connect individually with each satellite and provide a schedule as well as downlink data. It is more efficient to communicate with only one of the three satellites and obtain all the needed data as well as provide a schedule for

the entire constellation.[1] By uplinking a schedule for CASPER that includes activities for all three satellites, the latter can be accomplished.

Data Monitoring

The main objective of the 3CS project is to perform stereo imaging, resulting in large images to be transmitted to the ground. It is also necessary to provide the ground with the satellite's health at each pass. This will require much of the data to be transmitted to the master satellite before reaching a ground station.

The method that will be used for this task is inter-satellite communication resulting in the master obtaining the necessary data to transmit to the ground. First, the health will be obtained in a simple manner. Each satellite will carry four SCL databases onboard:

0 = default/persistent
1 = Petey
2 = Ralphie
3 = Sparky

Petey, Ralphie, and Sparky are the names of the three satellites after the mascots of the three involved universities. The 0 database will be used to store values upon startup in case the flight computer is re-started during flight. Each telemetry sample onboard a given satellite will fill that satellite's database. However, the ground will need to see each satellite's health. Therefore, the master satellite will perform an activity that tells the slave satellites to send their health packets to the master. Once the master receives the packets, it will be placed into the respective satellite's database onboard the master.

The same will be done with the images taken by the slave satellites. A ranking process will take place after the image is taken. The master may then ask for any number of things. It may ask for the most recent image (in the case of a stereo image being taken), the best ranked image, or any of the image slots where the images are stored. Once the master has received the desired images from the slaves, the master is ready to downlink them to a ground station.

3.4.3 SCL Model

The requirements faced in developing the SCL model were similar to those of the entire mission. These requirements are: monitor the health and status of the constellation; provide a user-oriented framework; and support the migration of autonomy.[2] The creation of the SCL model can be described in three sections as the approach, the development, and the testing.

Approach

The approach to the SCL model was to create a model that could be used aboard each of the satellites without the need to tailor it to each individual satellite. It was already mentioned that each satellite would carry, all four databases. This allows each satellite to maintain knowledge of the constellations resources and states. In addition to the four databases, the

master will need to control the other satellites. Another difficulty with creating a single model is the need to rotate the master satellite. Due to power constraints aboard the satellites the satellite dubbed master will need to change during flight. To create this model, a hierarchy of scripts was designed to accommodate several states of the satellites.

Development

Model development started with the design of the script hierarchy. Three main levels of scripts exist in the model. Figure 17 shows the scripting hierarchy for the SCL model.

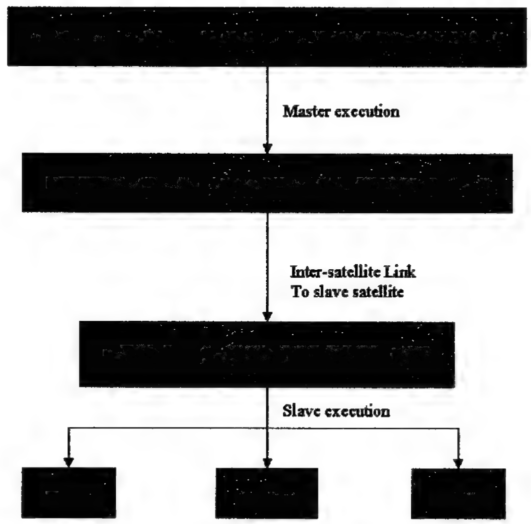


***Figure** 17. Hierarchy of scripts in SCL model.*

The hierarchy begins with scripts that are executed upon the commitment of a CASPER activity. These scripts are used by CASPER to analyze resources and provide a link between CASPER and the model developed by the EEDS team. Most of the CASPER scripts perform only the execution of another script level called the scheduling scripts. These scripts were provided mnemonics beginning with a schedule based on how they perform, schedule_send_hs for example. These scripts do not communicate directly to satellite hardware. Scheduling scripts perform inter-satellite communication to request the execution of local scripts. Scheduling scripts are only executed on the master satellite and they initiate the activities performed by the constellation.

After the ISL is successful, execution of the local scripts takes place. The local scripts are the scripts that perform satellite activities. Local scripts talk directly with the hardware and relay data back to the master satellite if desired. By providing the hierarchy of scripts in a single SCL model, all three satellites are capable of acting as master or slave.

Testing

Testing the SCL model and the flight software is a difficult task. The flight hardware was delivered to the AFRL for functional and environmental testing in late January. After

January, availability to test software on the satellites was no longer an option. Therefore, testbeds were created at the University of Colorado as a mock up of the actual satellites. Each testbed includes a PowerPC flight computer and interface board, EPS unit, imaging hardware, and communication hardware as well as an Ethernet connection and external power source, figure 18. The testbeds, however, do not include solar panels or batteries. While the test environment has not been perfect, it has provided a sound environment for functional testing and completion of the flight software. Three testbed satellites are used for supporting tests of inter-satellite communication and communication with a ground station.

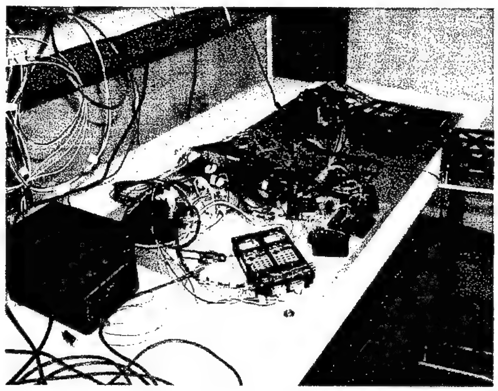


Figure 18. *Satellite hardware laid out as a testbed.*

3.4.4 In-flight Safeguarding

Since communication time with the satellites is limited, it is necessary to provide onboard safeguarding during flight. For 3CS, this is accomplished through the use of SCL. SCL provides the ability to monitor the satellite health and status, apply rules for various subsystems, and analyze derived data onboard the satellites. The following discussion outlines the methods used to safeguard the 3CS Constellation.

Health and Status

Each of the three satellites in the constellation carries 56 sensors that provide a basic telemetry sample. Included are current, voltage, and temperature sensors that provide information about the satellite. Every time a telemetry sample is taken, the recorded data is stored into the SCL database, as both the raw value and the more useful engineering unit.

However, the health and status includes much more than just the analog sensor readings. SCL rules allow for derived data to be obtained from the analog data. Based off the current from the solar panels, the satellite is determined to be in day or night, or based off the voltage to the cameras, the usage of the camera is determined as (0 off or 1 on). Derived data is most useful in creating constraints on the SCL scripts. For 3CS, an SCL constraint consists of a function that returns a "yes" or "no". The function may analyze several analog sensors and derived database items. The function is then evaluated in a script and based off the output, the script will either run or end. This is useful in preventing poorly scheduled activities or squabbling amongst the satellites.

Part of the mission requires downlinking health and status to remote HAM operators who will in turn post the data on the Internet for 3CS operators to view. Only the master satellite is allowed to downlink the health and status to prevent radio interference. Therefore, a constraint checks if the satellite is the master before downlinking to a remote ground station. The use of analog sensors, rules to create derived data, and constraints to prevent problems provides added insurance against satellite faults.

Error Detection

Another use of SCL rules is to detect and respond to failure modes effects and analysis (FMEA). Based on readings, either analog or derived, a rule can detect the possible error and act accordingly. Some FMEAs are fairly simple and require only turning off a piece of hardware or changing the master satellite. However, others are rather complex and require analysis on the ground by an operator. SCL rules have the ability to detect an error and record the necessary data that may be useful for an operator. An SCL rule may then find during a downlink that an error log exists and set the priority of downlinking the error log above downlinking the current health and status or an image. This will allow operators on the ground to act quickly in determining a quick resolution.

Flight Software – SCL

A distinct line lies between the low-level flight software that talks to the hardware and the higher level scripting of SCL. To bridge this difference a method of communicating between the two is being used to perform more efficiently. In order for SCL to obtain data from the FSW a decommutation record must exist in the SCL database. Each decommutation record carries a mnemonic used by SCL in scripts or rules. To perform more efficiently, the FSW carries many decommutation records associated with tasks. While an SCL script is executing, it sends several commands to the hardware. With each command the FSW replies with a decommutation record that SCL will analyze before sending the next command. If the decommutation reports a failure in sending the command, SCL will try several times. This allows SCL to know how the parts of each subsystem are running. If a command continuously fails, SCL may create an error log that may be reviewed by an operator. The link between SCL and the FSW is necessary in preventing two scripts from using the same hardware at the same time. By preventing this fault, more objectives will be met and fewer system errors will arise.

3.4.5 Post-Constellation Operations

Three Corner Satellite has two different mission periods. The three satellites will start as a constellation with formation flight and distributed tasking. The constellation will be drifting apart during the mission. Once the satellites drift beyond 100km apart they will lose communication amongst each other.

The secondary mission phase is a period where all three satellites are individual satellites performing the same mission objectives, but with no knowledge of the other satellites. This

period is known as the post-constellation period. The transition from a constellation to a post-constellation flight includes several factors. First, each satellite will essentially become a master satellite and run its own schedule. Secondly, operators hope to be able to communicate with each satellite individually during ground station passes.

Post-constellation poses several problems such as downlinking to the ground. When downlinking to remote ground satellites none of the satellites will know what the others are doing. It will become more important to provide a schedule with fewer conflicts during this period. In addition, SCL rules and constraints that concern ISL will be deactivated. Lastly, SCL scripts at the scheduling level will no longer be used, as the ISL to other satellites will not be necessary. Post-constellation mission objectives will include stereo imaging by taking two images separated by an interval and the use of CASPER scheduling software.

4.0 Personnel Supported

Faculty:

Dave Beckwith
Brian Egaas
Elaine Hansen
Steve Wichman

Students:

Shannon Atencio
Israel Archuleta
Aleksander Arustamyan
Nicholas Blackwell
Casey Brewer
Matt Colgan
Steve Crouch
Matt Culbreth
Lauren Dermyer
Benjamin Drury
Stephan Esterhuizen
Matt Gadda
Chris Gilmer
Matthew Iwaskow
Peter Kimball
Otto Krauss
Steve Levin-Stankevich
Jason Lonergan
James Anthony Lowrey
Jonathan Mabra
Alexander McClearn
Kenneth Mentasti
Jennifer Michels
Ben Mottinger
Kiran Murthy
Jesse Negrete
Whitney Noon
Ryan Olds
Alvin Ortiviz
Eric Pahl
Erica Raine
Angel Rocha
Victoria Scarffe
Aaron Shalosky
Alexander Shenkin
Thomas Steen
Andrew Taylor
Matthew Vellone
Michael Walters

5.0 PUBLICATIONS & PRESENTATIONS

Culbreth, Matt. "Broadcast Communications", http://spacegrant.colorado.edu/symposium/final/Matt_Culbreth.doc, 2002 Space Grant Student Symposium, Boulder, CO, 2002

Olds, Ryan. "3CS Imaging", http://spacegrant.colorado.edu/symposium/final/Ryan_Olds.doc, 2002 Space Grant Student Symposium, Boulder, CO 2002

Krauss, Ervin. "Slowdown Mechanism", http://spacegrant.colorado.edu/symposium/final/Ervin_Krauss.doc, 2002 Space Grant Student Symposium, Boulder, CO 2002

Blackwell, Nick. "Historical Perspective on Mission Operations", http://spacegrant.colorado.edu/symposium/final/Nick_Blackwell.doc, 2002 Space Grant Student Symposium, Boulder, CO 2002

Esterhuizen, Stephan. "Solar Panel Emulator", http://spacegrant.colorado.edu/symposium/final/Stephan_Esterhuizen.doc, 2002 Space Grant Student Symposium, Boulder, CO 2002

Mayer, Dan. "Intersatellite Communication", 2002 Space Grant Student Symposium, Boulder, CO 2002

Levin-Stankevich, Stephen. "Commanding and Safeguarding the Three Corner Satellite Constellation", http://spacegrant.colorado.edu/boulder/published_papers/Stankevich_SmallSat.doc, 16th Annual AIAA/USU Conference on Small Satellites, Logan, Utah, 2001

"Better, Faster, and Cheaper Missions Through Automated Planning and Scheduling," Elaine Hansen, Colette Wilklow, and Shankini Doraisingam, World Automation Congress 2000, Maui, Hawaii.

"Three Corner Sat Constellation: C&DH, Stereoscopic Imaging, and End-to-End Data System," Elaine Hansen and Tony Colaprete, and Dan Rodier, University of Colorado at Boulder, Stephen Horan and Bobby Anderson, New Mexico State University, Brian Underhill, Assi Friedman, Joyce Wong, and Helen Reed, Arizona State University, Proceedings of the 13th Annual Conference on Small Satellites, Logan, Utah August 23-26, 1999.

"Three Corner Sat Constellation: Management, Systems, Spacecraft Bus, Micropropulsion/ Payloads," Brian Underhill, Assi Friedman, Joyce Wong, and Helen Reed, Arizona State University; Elaine Hansen, Tony Colaprete, and Dan Rodier, University of Colorado at Boulder; Stephen Horan and Bobby Anderson, New Mexico State University, Proceedings of the 13th Annual AUAA/Utah State University Conference on Small Satellites, Logan, Utah August 1999.

"Three Corner Sat Constellation: Communication to Support Formation Flying," B. Anderson and S. Horan, New Mexico State University; B. Underhill, A. Friedman, J. Wong, and H. Reed, Arizona State University, E. Hansen and D. Rodier, University of Colorado at Boulder, Proceedings of the 13th Annual AUAA/Utah State University Conference on Small Satellites, Logan Utah. August 23-26, 1999.

6.0 INTERACTIONS & TRANSITIONS

6.1 Consultive and Advisory Functions

Elaine Hansen is serving as a member of the USRA Space Technology Council where she advised USRA on educational needs and opportunities. Elaine is also serving as a member of the student space research program which is part of the National Space Grant Directors Council.

6.2 Transitions

There is interest by several groups in the potential technology applications in the systems developed by student at CU.. These include:

1. Demonstration of autonomous scheduler and planner in flight and ground systems.
   - JPL
   - Dr. Steve Chein
2. Coordinated operation of a constellation of satellites
   - Lockheed Martin
   - Dr. Suraj Rawal
3. Demonstration of Constellation Operations
   - JPL and AFRL
   - Dr. Steve Chien and Jeff Ganley
4. Development and Demonstration of Slow Down Mechanism
   - Planetary Systems
   - Dr. Walter Holemans

7.0 References

[Unde 99] Underhill, B., et al. "Three Corner Satellite Constellation - Arizona State University: Management; Electrical Power System; Structures, Mechanism, Thermal and Radiation; Attitude/Orbit Determination and Control; ASU Micropropulsion Experiment; and Integration," *Proceedings of the 13th Annual AIAA/USU Conference on Small Satellites,* SSC99-III-1, Logan, UT, August, 1999.

[Hans 99] Hansen, E., et al. "Three Corner Sat Constellation: C&DH, Stereoscopic Imaging, and End-to-End Data System," *Proceedings of the 13th Annual AIAA/USU Conference on Small Satellites*, SSC99-IV-1, Logan, UT, August, 1999.

[Hora 99] Horan, S., et al., "Three Corner Satellite Constellation - New Mexico State University: Communications, LEO Telecommunications Services, Intersatellite Communications, and Ground Stations and Network," *Proceedings of the 13th Annual AIAA/USU Conference on Small Satellites,* SSC99-VI-7, Logan, UT, August, 1999.

[NAS 99] National Aeronautics and Space Administration, *Hitchhiker Customer Accommodations & Requirements Specifications*, 740-SPEC-008, Goddard Space Flight Center, 1999.

[Hor 00] Horan, S., "3 Corner Satellite Communications Link Estimates," UN2-RPT-31114-403, Rev.a, Las Cruces, NM, 2000.

[Hor 01a] Horan, S., Alvarez, L., and Silva, A., "Assembly Procedure - Dual Transceiver," UN2-PLN-31114-417, Rev. a, Las Cruces, NM, 2001.

[Hor 01b] Horan, S., Alvarez, L., Jourdan, M., and Silva, A., "Manufacturers' Acceptance Test Procedure," UN2-RPT-31114-423, Rev.a, Las Cruces, NM 2001.

[Hor 02] Horan, S., Alvarez, L., Anderson, B., Pachecho, R., and Jourdan, M., "Dual Transceiver User's Manual," UN2-MAN-31114-404, Rev.B, Las Cruces, NM, 2002.

[Mau 02] K. Mauldin, "Networking Satellite Ground Stations Using LabVIEW," *Proc. International Telemetering Conference*, 02-10-4, San Diego, CA, October 2002.

[1] Zetocha, Paul: Satellite Cluster Command and Control, Http://www.interfacecontrol.com/papers/z2_0106.pdf

[2] Rowland, V. Derock, C.: Flight Software Design Requirements, *Three Corner Satellite Constellation*, http://spacegrant.colorado.edu/~swichman/3cs/requirements_top_flight.html